RECENT PROGRESS IN THE SPHINX SPEECH RECOGNITION SYSTEM

Kai-Fu Lee, Hsiao-Wuen Hon, Mei-Yuh Hwang
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This paper describes recent improvements in the SPHINX Speech Recognition System. These enhancements include function-phrase modeling, between-word coarticulation modeling, and corrective training. On the DARPA resource management task, SPHINX attained a speaker-independent word accuracy of 96% with a grammar (perplexity 60), and 82% without grammar (perplexity 997).

1. Introduction

SPHINX is a large-vocabulary, speaker-independent, continuous speech recognition system based on discrete hidden Markov models (HMMs) with LPC-derived parameters. In order to deal with the problem of speaker independence, we added knowledge to these HMMs in several ways. We represented additional knowledge through the use of multiple codebooks. We also enhanced the recognizer with word duration modeling. In order to model co-articulation in continuous speech, we introduced the use of function-word-dependent phone models, and generalized triphone models.

More recently, we have made considerable progress with the SPHINX System. We reformulated the generalized triphone clustering algorithm as a maximum-likelihood procedure, and carried out some experiments with generalized triphones. We also implemented and evaluated the modeling of function phrases, and between-word coarticulation modeling using generalized triphones. The latter experiment reduced SPHINX's error rate by 24-44%. We modified the corrective training algorithm [1] for speaker-independent, continuous speech recognition. Corrective training reduced SPHINX's error rate by 20-24%.

In this paper, we will describe all components of the SPHINX System, with emphasis on the recent improvements. The SPHINX System has been described in [2] and [3]. Publications on the recent improvements will be forthcoming.

On the 991-word DARPA resource management task, SPHINX achieved speaker-independent word recognition accuracies of 82% and 96%, with grammars of perplexity 991 and 60, respectively. Results with the 1988 and 1989 test data resulted in 78 and 76% without grammar, and 96% and 94% with the word pair grammar.

2. Speech Representation

The speech is sampled at 16 KHz, and pre-emphasized with a filter of $1 - 0.97z^{-1}$. Then, a Hamming window with a width of 20 msec is applied every 10 msec. Autocorrelation analysis with order 14 is followed by LPC analysis with order 14. Finally, 12 LPC-derived cepstral coefficients are computed from the LPC coefficients, and these LPC cepstral coefficients are transformed to a mel-scale using a bilinear transform.

These 12 coefficients are vector quantized into a codebook of 256 prototype vectors. In order to incorporate additional speech parameters, we created two additional codebooks. One codebook is vector quantized from *differential coefficients*. The differential coefficient of frame n is the difference between the coefficient of frame n+2 and frame n–2. This 40 msec. difference captures the slope of the spectral envelope. The other codebook is vector quantized from *energy* and *differential energy* values.

3. Context-Independent HMM Training

SPHINX is based on phonetic hidden Markov models. We identified a set of 48 phones, and a hidden Markov model is trained for each phone. Each phonetic HMM contains three discrete output distributions of VQ symbols. Each distribution is the joint density of the three codebook pdf's, which are assumed to be independent. The use of multiple codebooks was introduced by Gupta, *et al.* [4].

We initialize our training procedure with the TIMIT

phonetically labeled database. With this initialization, we use the forward-backward algorithm to train the parameters of the 48 phonetic HMMs. The training corpus consists of 4200 task-domain sentences spoken by 105 speakers. For each sentence, word HMMs are constructed by concatenating phone HMMs. These word HMMs are then concatenated into a large sentence HMM, and trained on the corresponding speech. Because the initial estimates are quite good, only two iterations of the forward-backward algorithm are run. This training phase produces 48 *context-independent* phone models. In the next two sections, we will discuss the second training phase for *context-dependent* phone models.

4. Function Word/Phrase Dependent Models

One problem with continuous speech is the unclear articulation of function words, such as *a*, *the*, *in*, *of*, etc. Since the set of function words in English is limited and function words occur frequently, it is possible to model each phone in each function word separately. By explicitly modeling the most difficult sub-vocabulary, recognition rate can be increased substantially. We selected a set of 42 function words, which contained 105 phones. We modeled each of these phones separately.

We have found that function words are hardest to recognize when they occur in clusters, such as *that are in the*. The words are even less clearly articulated, and have strong inter-word coarticulatory effects. In view of this, we created a set of phone models specific to *function phrases*, which are phrases that consist of only function words. We identified 12 such phrases, modified the pronunciations of these phrases according to phonological rules, and modeled the phones in them separately. A few examples of these phrases are: *is the*, *that are*, and *of the*.

5. Generalized Triphone Models

The function-word and function-phrase dependent phone models provide better representations of the function words. However, simple phone models for the non-function words are inadequate, because the realization of a phone crucially depends on context. In order to model the most prominent contextual effect, Schwartz, *et al.* [5] proposed the use of *triphone models*. A different triphone model is used for each left and right context. While triphone models are sensitive to neighboring phonetic contexts, and have led to good results, there are a very large number of them, which can only be sparsely trained. Moreover, they do not take into account the similarity of certain phones in their affect on other phones (such as /b/ and /p/ on vowels).

In view of this, we introduce the *generalized triphone model*. Generalized triphones are created from triphone models using a clustering procedure:

1. An HMM is generated for every triphone context.
2. Clusters of triphones are created; initially, each clusters consists of one triphone.
3. Find the *most similar* pair of clusters which represent the same phone, and merge them.
4. For each pair of same-phone clusters, consider moving every element from one to the other.
    1. Move the element if the resulting configuration is an improvement.
    2. Repeat until no such moves are left.
5. Until some convergence criterion is met, go to step 2.

To determine the distance between two models, we use the following distance metric:

$$D(a,b) = \frac{(\prod_i (P_a(i))^{N_a(i)}) \cdot (\prod_i (P_b(i))^{N_b(i)})}{\prod_i (P_m(i))^{N_m(i)}} \quad (1)$$

where $D(a,b)$ is the distance between two models of the same phone in context a and b. $P_a(i)$ is the output probability of codeword i in model a, and $N_a(i)$ is the count of codeword i in model a. m is the merged model by adding N_a and N_b. In measuring the distance between the two models, we only consider the output probabilities, and ignore the transition probabilities, which are of secondary importance.

Equation 1 measures the ratio between the probability that the individual distributions generated the training data and the probability that the combined distribution generated the training data. Thus, it is consistent with the maximum-likelihood criterion used in the forward-backward algorithm. This distance metric is equivalent to, and was motivated by, entropy clustering used in [6] and [7].

This context generalization algorithm provides the ideal means for finding the equilibrium between trainability and sensitivity. Given a fixed amount of

training data, it is possible to find the largest number of trainable detailed models. Armed with this technique, we could attack any problem and find the "right" number of models that are as sensitive and trainable as possible. This is illustrated in Figure 1, which shows that the optimal number of models increases as the training data is increased.

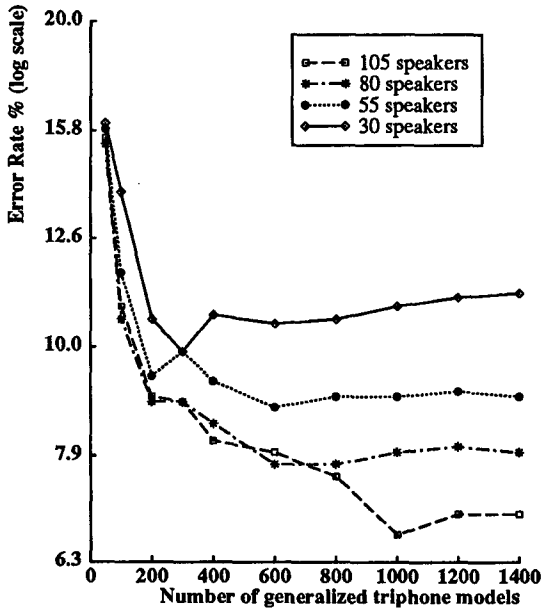

Figure 1: Error rate as a function of the amount of training and the number of models.

6. Between-Word Coarticulation Modeling

Triphone and generalized triphone models are powerful subword modeling techniques because they account for the left and right phonetic contexts, which are the principal causes of phonetic variability. However, triphone-based models consider only intra-word context. For example, in the word *speech* (/s p iy ch/), both left and right contexts for /p/ and /iy/ are known, while the left context for /s/ and the right context for /ch/ are a special symbol for "word boundary". However, in continuous speech, a word-boundary phone is strongly affected by the phone beyond the word boundary. This is especially true for short function words like *the* or *a*.

A simple extension of triphones to model between-word coarticulation is problematic because the number of triphone models grows sharply when between-word triphones are considered. For example, there are 2381 within-word triphones in our 991-word task. But there are 7057 triphones when between-word triphones are also considered.

Therefore, generalized triphones are particularly suitable for modeling between-word coarticulation. We first generated 7057 triphone models that accounted for both intra-word and inter-word triphones. These 7057 models were then clustered into 1000 generalized triphone models. The membership of each generalized triphone is retained, so that inter-word contextual constraints can be applied during training and recognition.

The main change in the training algorithm is in the construction of the sentence model. Two connections are now needed to link two words together. The first uses the known context to connect the appropriate triphones, and the second allows for the possibility of a between-word silence. In that case, a silence context is used. Figure 2 illustrates the word boundary network of two words, where word w_1 consists of phones A, B, and C, and word w_2 consists of D, E, and F.

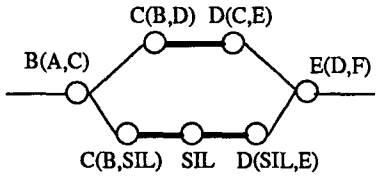

Figure 2: Sentence network connection during training. Here word w_1 consists of phones A, B, and C, and word w_2 consists of D, E, and F. P(L,R) represents a phone P with left-context phone L and right-context phone R.

For words with only one or two phones, sentence model concatenation is more complex. If w_2 is pronounced (D E), then both D(C,E) and D(SIL,E) must be further forked into E(D,X) and E(D,SIL), where X is the first phone of the next word. This is even more complicated when several one-phone and two-phone words are concatenated. To reduce the complexity of the pronunciation graph of a sentence, we introduce dummy states to merge transitions whose expected contexts are the same.

The recognition algorithm must be modified because words may now have multiple begining and ending phones. Figure 3 illustrates the connection between two words during recognition. Like the training phase, the two words are connected both directly and through a silence. If one or both of the triphones has not occurred in the training data, we use the context-independent phone (or monophone) instead. Therefore, the direct connection between two words could be embodied in one of four forms:

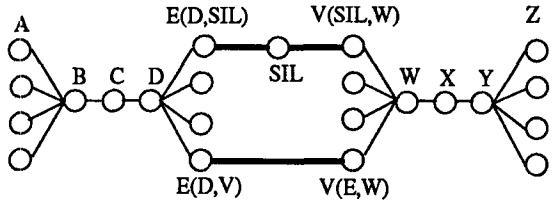

Figure 3: Transitioning from one word (A B C D E) to another (V W X Y Z) in recognition.

- triphone to triphone.
- triphone to monophone.
- monophone to triphone.
- monophone to monophone.

The modeling of between-word coarticulation reduced SPHINX's error rate by 24-44%, for different test sets and grammars. More details about our implementation and results can be found in [8].

7. Corrective Training

Bahl *et al.* [1] introduced the corrective training algorithm for HMMs as an alternative to the forward-backward algorithm. While the forward-backward algorithm attempts to increase the probability that the models generated the training data, corrective training attempts to maximize the recognition rate on the training data. This algorithm has two components: (1) *error-correction learning* — which improves correct words and suppresses misrecognized words, (2) *reinforcement learning* — which improves correct words and suppresses near-misses. Applied to the IBM speaker-dependent isolated-word office correspondence task, this algorithm reduced the error rate by 16% on test data and 88% on training data. This improvement, while significant, suggests that corrective training becomes overly specialized for the training data.

In this study, we extend the corrective and reinforcement learning algorithm to *speaker-independent, continuous speech* recognition. Speaker independence may present some problems, because corrections appropriate for one speaker may be inappropriate for another. However, with a speaker-independent task, it is possible to collect and use a large training set. More training provides not only improved generalization but also a greater coverage of the vocabulary. We also use *cross-validation* to increase the effective training data size. Cross-validation partitions the training data and determines misrecognitions using models trained on different partitions. This simulation of actual recognition leads to more realistic misrecognitions for error correction.

Extension to continuous speech is more problematic. With isolated-word input, both error-correcting and reinforcement training are relatively straighforward, since all errors are simple substitutions. Bahl, *et al.* [1] determined both misrecognized words and near-misses by matching the utterance against the entire vocabulary. However, with continuous speech, the errors include insertions and deletions. Moreover, many substitutions appear as phrase-substitutions, such as *home any* for *how many*. These problems make reinforcement learning difficult. We propose an algorithm that hypothesizes near-miss sentences for any given sentence. First, a dynamic programming algorithm is used to align each correct sentence with the corresponding misrecognized sentence in the cross-recognized training set to produce an ordered list of likely *phrase substitutions*. Since simple text-to-text alignment would not be sensitive to sub-word and sub-phone similarities, we used a frame-level distance metric. This list of phrase substitutions are then used to randomly hypothesize near-miss sentences for reinforcement learning.

Our experiments with corrective and reinforcement learning showed that our modifications led to a 20% error-rate reduction without grammar (72% on training set), and a 23% reduction with grammar (63% on training set). This demonstrated that increased training, both through speaker-independent data collection and through cross-validation, narrowed the gap between the results from training and testing data. Furthermore, this showed that our extension of the IBM corrective training algorithm to continuous speech was successful. More details about this work are described in [9] and [10].

8. Summary of Training Procedure

The SPHINX training procedure operates in three stages. In the first stage, 48 context-independent phonetic models are trained. In the second stage, the models from the first stage are used to initialize the training of context-dependent phone models, which could be generalized triphone models and/or the function word/phrase dependent models. Since many parameters in the context-dependent models were never observed, we interpolate the context-dependent model parameters with the corresponding context-independent ones. We use *deleted interpolation* [11] to derive appropriate weights in the interpolation. The third and final stage uses corrective training to refine the dis-

criminatory ability of the models. The SPHINX training procedure is shown in Figure 4.

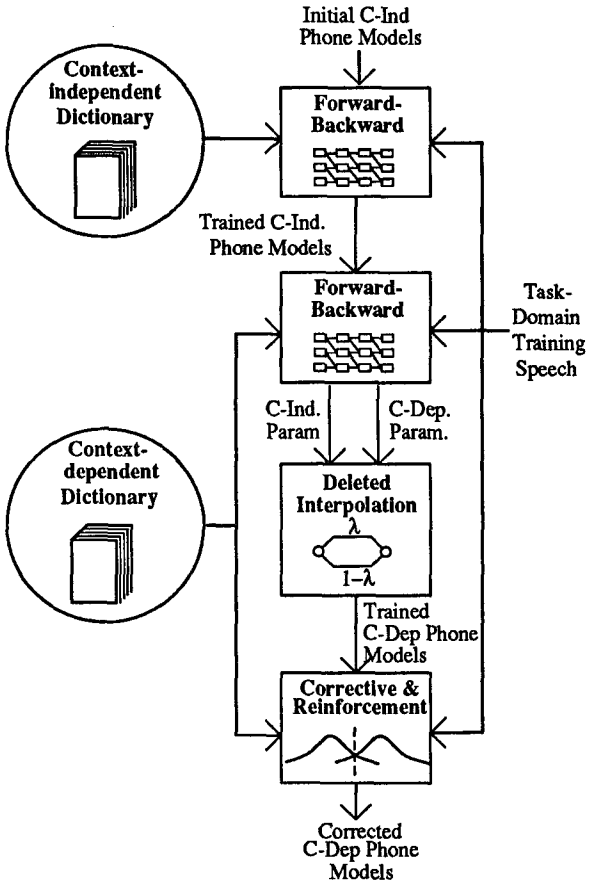

Figure 4: The SPHINX Training Procedure.

9. HMM Recognition with Word Duration

For recognition, we use a Viterbi search that finds the optimal state sequence in a large HMM network. At the highest level, this HMM is a network of word HMMs, arranged according to the grammar. Each word is instantiated with its phonetic pronunciation network, and each phone is instantiated with the corresponding phone model. Beam search is used to reduce the amount of computation.

One problem with HMMs is that they do not provide very good duration models. We incorporated word duration into SPHINX as a part of the Viterbi search. The duration of a word is modeled by a univariate Gaussian distribution, with the mean and variance estimated from a supervised Viterbi segmentation of the training set. By precomputing the duration score for various durations, this duration model has essentially no overhead.

10. Results

The SPHINX System was tested on 150 sentences from 15 speakers. These sentences were the official DARPA test data for evaluations in March and October 1987. The word accuracies for various versions of SPHINX with the word-pair grammar (perplexity 60) and the null grammar (perplexity 991) are shown in Table 1. Word accuracy is defined as the percent of words correct minus the percent of insertions.

Version	No Grammar	Word Pair
1 Codebook	25.8%	58.1%
3 Codebooks	45.3%	84.4%
+Duration	49.6%	83.8%
+Fn-word	57.0%	87.9%
+Fn-phrase	59.2%	88.4%
+Gen-triphone	72.8%	94.2%
+Between-word	77.9%	95.5%
+Corrective	81.9%	96.2%

Table 1: Results of various versions of SPHINX.

The first improvement was obtained by adding additonal feature sets and codebooks. Next, we found duration modeling to be helpful when no grammar was used. Modeling function words and generalized triphones both led to substantial improvements. We also found that generalized triphones outperformed triphones, while saving 60% memory*. The improvements from function-phrase dependent modeling encouraged us to implement between-word triphone models. This led to substantial improvements with no increase in the number of models. Finally, we showed the effectiveness of our extension of the corrective training algorithm to speaker-independent continuous speech.

Since the above experiments were repeatedly run on the same set of test data, it is important to verify that SPHINX is capable of achieving comparable levels of performance on new test data. Recently, SPHINX was evaluated on two new sets of test data (June 1988 evaluation and February 1989 evaluation). With no grammar, recognition accuracies of 78.1% and 76.4% were obtained on these two test sets. With the word-pair grammar, the accuracies were 95.7% and 93.9%.

*More detailed descriptions and results on contextual modeling can be found in [2] or [3].

11. Conclusion

This paper has presented an up-to-date description of the SPHINX Speech Recognition System. We have described a number of recent improvements, including function-phrase modeling, between-word coarticulation modeling, and corrective and reinforcement training.

Through these techniques we demonstrated that accurate large-vocabulary speaker-independent continuous speech recognition is feasible. We report recognition accuracies of 82% and 96% with grammars of perplexity 997 and 60. The results degraded somewhat on new test data, but remain highly accurate. These results were made possible by three important factors: (1) ample training data, (2) a powerful learning paradigm, and (3) knowledge-guided detailed models.

Encouraged by these results, we will continue in the current SPHINX framework, and direct our future efforts to improving each of these three areas. We feel that work in each of the three directions will lead to substantial progress, and hope that our future work will contribute to the next generation of accurate, robust, and versatile speech recognition systems.

Acknowledgments

The authors wish to thank the CMU Speech Group for their support and contributions. This research was partly sponsored by Defense Advanced Research Projects Agency Contract N00039-85-C-0163, and partly by a National Science Foundation graduate fellowship.

References

1. Bahl. L.R., Brown, P.F., De Souza, P.V., Mercer, R.L., "A New Algorithm for the Estimation of Hidden Markov Model Parameters", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1988.

2. Lee, K.F., *Large-Vocabulary Speaker-Independent Continuous Speech Recognition: The SPHINX System*, PhD dissertation, Computer Science Department, Carnegie Mellon University, April 1988.

3. Lee, K.F., *Automatic Speech Recognition: The Development of the SPHINX System*, Kluwer Academic Publishers, Boston, 1989.

4. Gupta, V.N., Lennig, M., Mermelstein, P., "Integration of Acoustic Information in a Large Vocabulary Word Recognizer", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1987, pp. 697-700.

5. Schwartz, R., Chow, Y., Kimball, O., Roucos, S., Krasner, M., Makhoul, J., "Context-Dependent Modeling for Acoustic-Phonetic Recognition of Continuous Speech", *IEEE International Conference on Acoustics, Speech, and Signal Processing*, April 1985.

6. Lucassen, J.M., "Discovering Phonemic Baseforms: an Information Theoretic Approach", Research Report RC 9833, IBM, February 1983.

7. Brown, P., *The Acoustic-Modeling Problem in Automatic Speech Recognition*, PhD dissertation, Computer Science Department, Carnegie Mellon University, May 1987.

8. Hwang, M.Y., Hon, H.W., Lee, K.F., "Between-Word Coarticulation Modeling for Continuous Speech Recognition", Technical Report, Carnegie Mellon University, March 1989.

9. Lee, K.F., Mahajan, S., "Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition", Technical Report CMU-CS-89-100, Carnegie Mellon University, January 1989.

10. Lee, K.F., Mahajan, S., "Corrective and Reinforcement Learning for Speaker-Independent Continuous Speech Recognition", Submitted to Computer Speech and Language.

11. Jelinek, F., Mercer, R.L., "Interpolated Estimation of Markov Source Parameters from Sparse Data", in *Pattern Recognition in Practice*, E.S. Gelsema and L.N. Kanal, ed., North-Holland Publishing Company, Amsterdam, the Netherlands, 1980, pp. 381-397.